Mechatronic Systems Engineering
School of Engineering Science
SIMON FRASER UNIVERSITY

Mechatronic Systems Engineering
School of Engineering Science
SIMON FRASER UNIVERSITY

MSE 450 – Real-Time and Embedded Control Systems
- Project Report -

3 Phase Motor Control

Due April 12th, 2015

I hereby certify project work was equal amongst all group members

Kevin Mathiasen	301162339	______________________
Mavin Bautista	301175093	______________________
Harry He	301161400	______________________

Table of Contents

1. Introduction

The purpose of this project was to control the angular velocity of a 3 phase brushless DC motor with a gyroscope. Input was taken from the gyroscope, and based on that data a PID controller was used to continuously change the power supplied to the motor, in order to control the speed.

2. High Level Design

2.1. Requirements Form

Name	3 Phase Brushless DC Motor Position Control
Purpose	To control the angular velocity of the motor's stator based on the input of a gyroscope.
Inputs	Start Button Reset Button 3x Hall Effect Sensor signals Encoder Signals Gyroscope angular position PID controller Gains
Outputs	Motor Stator angular velocity
Functions	Capture gyroscope data to determine desired angular velocity Set the motor's velocity based on the gyroscope
Performance	Accuracy: +/- 0.05 degrees Rise time: 0.005 sec Settling Time: 0.001 sec
Manufacturing Cost	STM32f4 Discover MCU = $20 BLDC Motor with built in Hall Effect Sensors and encoders = $200 STL6324 3 phase motor driver IC = $10
Power	5 V DC power supply on board
Physical (size/weight)	5 lbs, 6x6x2 in

Table 1: *Requirements form*

2.2. Graphical Views

The highest level design of our system is shown below in Figure 1. The later sections of this report will also expand upon, and give a more detailed view of a select number of classes.

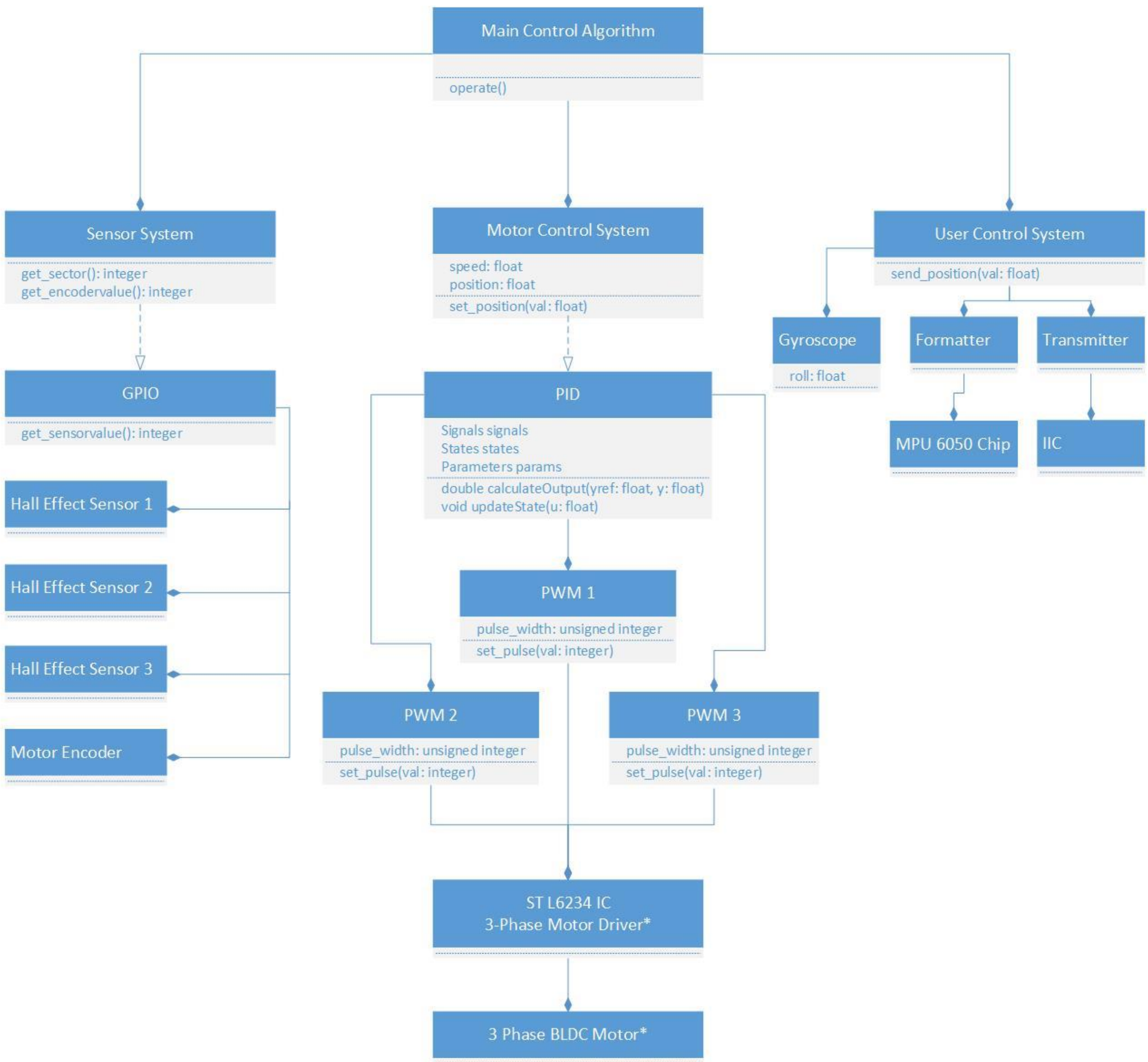


***Figure 1:** High level graphical view of the system*

3. Hardware Components

3.1. The STM32F407 Microcontroller Unit (MCU)

For this project, the MCU used is the STM32F407 Discovery MCU. It is an MCU manufactured by ST Microelectronics and uses an ARM Cortex M4 core. Below is a figure showing the board as well as its key features:

Features/Specs:

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM in an LQFP100 package
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone ST-LINK/V2 (with SWD connector for programming and debugging)
- Board power supply: through USB bus or from an external 5 V supply voltage
- External application power supply: 3 V and 5 V
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02, ST MEMS audio sensor, omni-directional digital microphone
- CS43L22, audio DAC with integrated class D speaker driver
- Eight LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power on
  - Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  - 2 USB OTG LEDs LD7 (green) VBus and LD8 (red) over-current
- Two push buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing

Figure 2: *STM32F407 MCU schematic*

In order to program our MCU, the Uvision 4 Integrated Development Environment (manufactured by Keil) was used. Keil Uvision 4 fully supports the ARM programming model and thus STM32F4 discover series of microcontrollers. All software components of this project was produced in this aforementioned IDE.

3.2. 3-Phase Brushless DC Motor

For this project, an EC-max 30, 40 Watt, 3 phase brushless DC motor is to be controlled by the MPU-6050 MEMS 3-Axis Gyroscope/Accelerometer interfacing with the STM32F407 discovery MCU.

Brushless DC Motors have permanent magnet rotors, and as such the polarities cannot be switched. Instead, commutation is achieved by switching the current on the stators, based on the position of the rotor. The typical configuration of a BLDC motor is shown below in Figure 3.

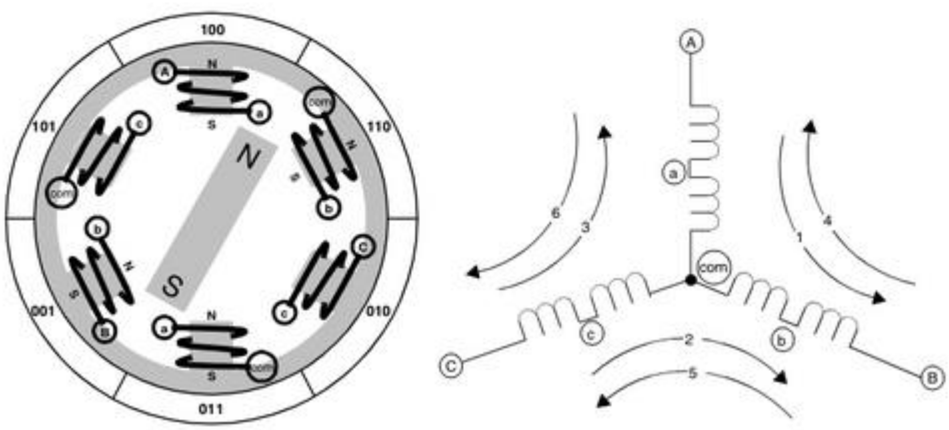


Figure 3: *Brushless DC motor configuration [6]*

The current to the stators is provided by three separate pulse width modulated sine signals, which are out of phase by 120 degrees, hence the name 3 phase motor [5]. The signal is generated using a lookup table, and is shown below in Figure 4.

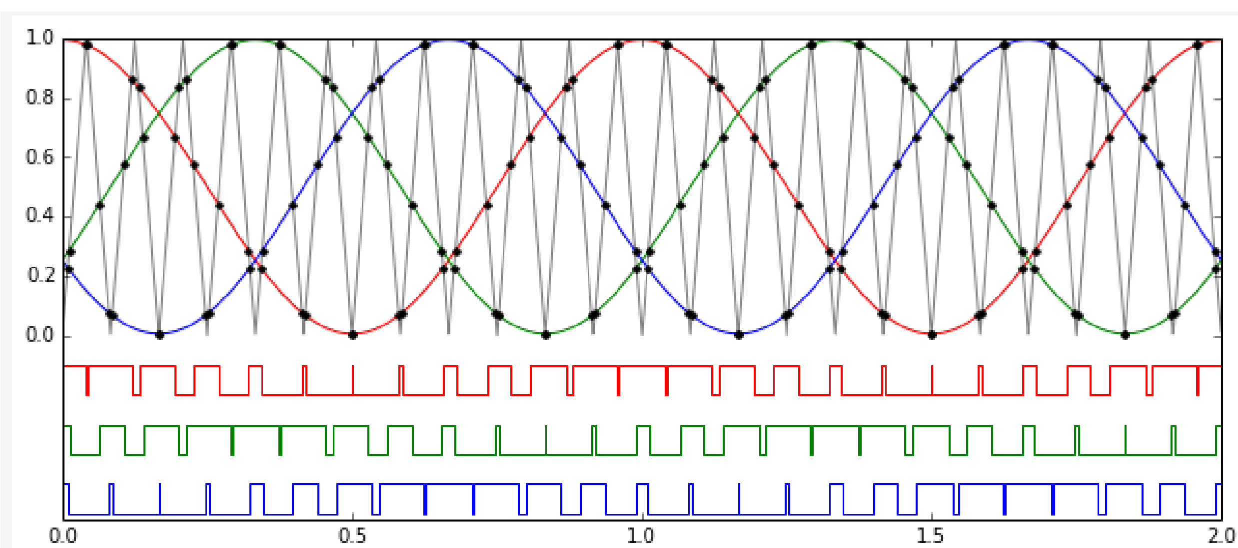


Figure 4: 3 sine signals used to control the BLDC motor

The motor contains three Hall Effect sensors, A, B, and C, in order to determine which phase the rotor is currently in. These 3 sensors provide six states, shown below in Figure 5, which can be used to determine which sector the rotor is in [4].

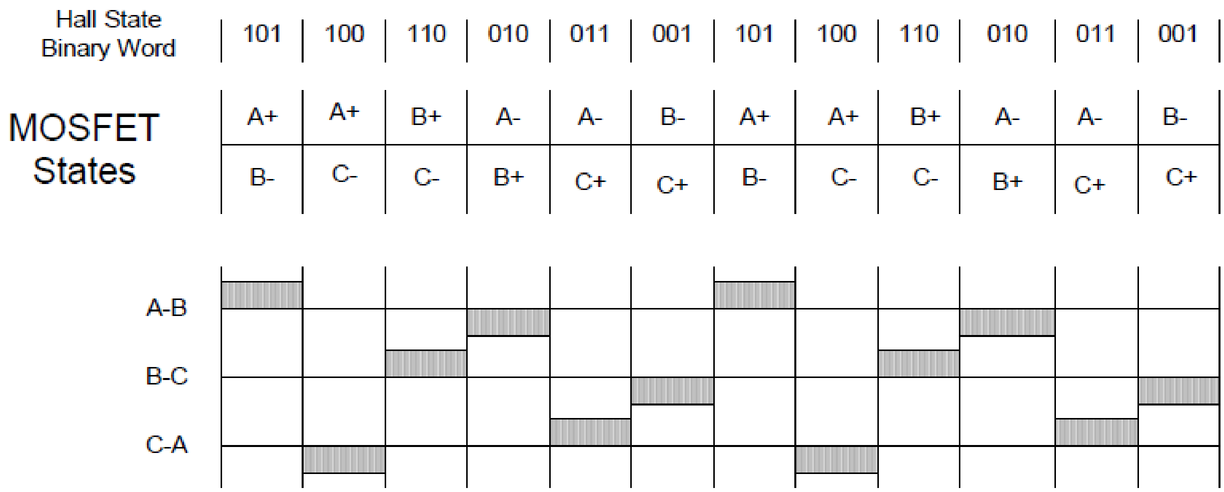


Figure 5: *Hall Effect, and MOSFET states of the BLDC motor [4].*

Based on these states, the MCU calculates the appropriate enable signals to be sent to the circuit.

3.3. MEMS 3-Axis Gyroscope and Accelerometer

The MPU-6050 3 axis accelerometer and gyroscope was used in order to control the three phase motor. The gyroscope uses the Inter Integrated Circuit (IIC) interface in order to communicate with our microcontroller. The IIC protocol is a universal protocol for all IIC devices, and is show below in Figure 6.

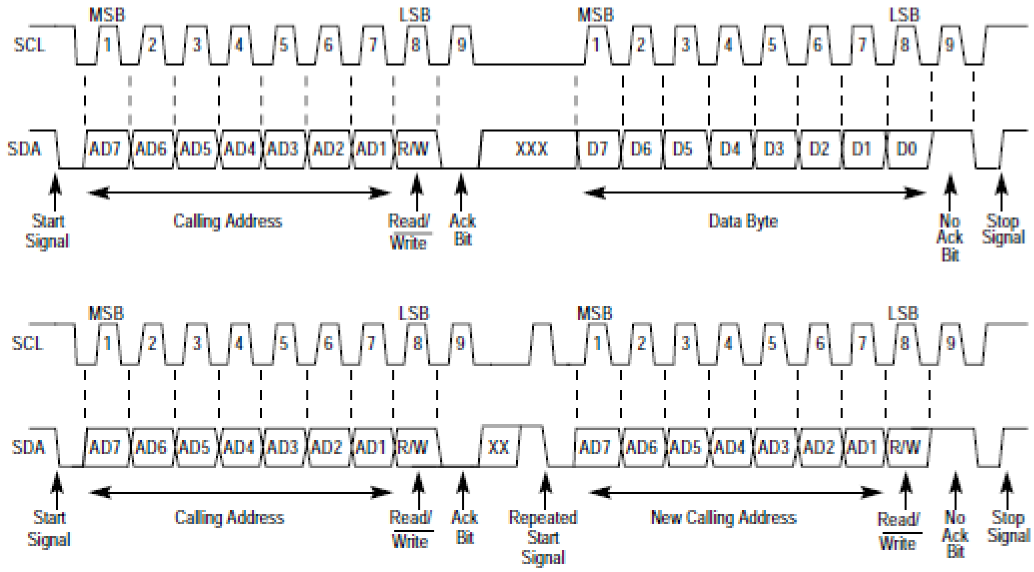


Figure 6: *IIC communication structure. (Top) data sequence. (Bottom) address sequence [2].*

The sequence required to read data from the gyroscope, as well as common IIC terms is shown below in Figure 7.

Burst Read Sequence

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

I²C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I²C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9th clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9th clock cycle
RA	MPU-30X0 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high

Figure 7: *IIC read sequence for MPU-6050 [3].*

3.4. Motor Driver Circuit

Due to the output voltage limitations of the ST32F4 discover MCU (max 5V supply) an intermediate interfacing module must be used to allow the MCU to control the 3 phase motor. In addition to the need of voltage amplification and an external power source, the signals from the MCU must be translated and filtered to allow adequate operation of the 3 phase motor. In order to achieve this, we implemented the L6324 3 phase motor driver integrated circuit (IC) manufactured by ST Microelectronics. The following figure is an excerpt from the IC's corresponding data sheet that illustrates how the circuit is connected.

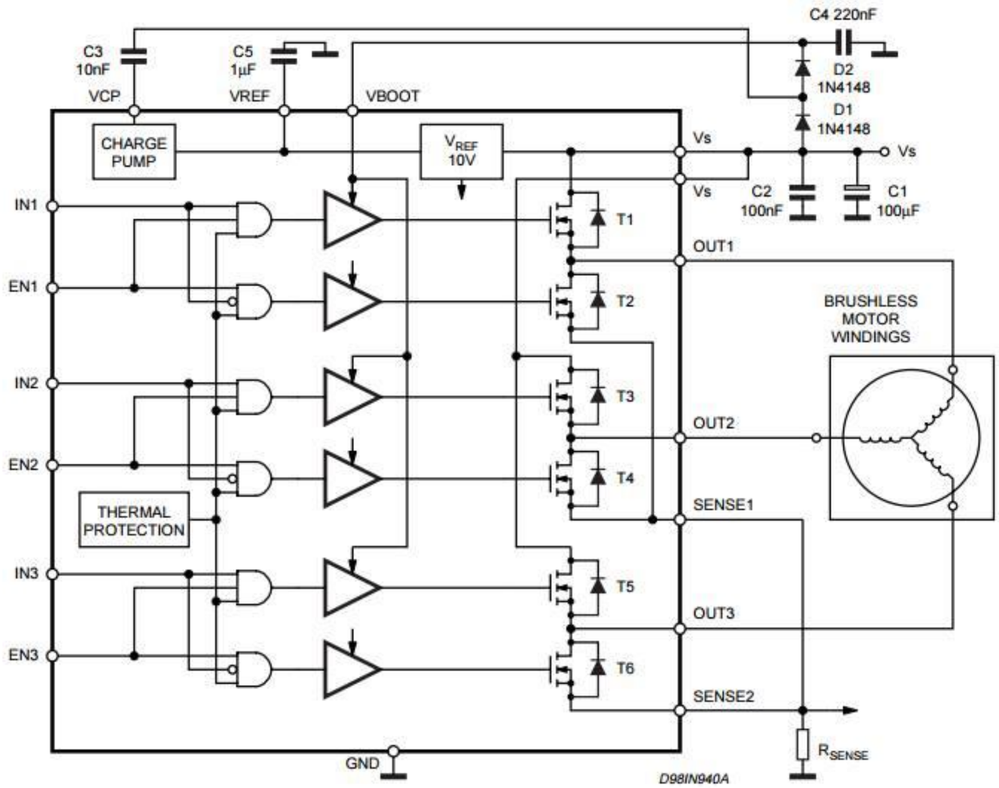


Figure 8: *Circuit diagram of 3 phase motor driver [7].*

4. Controller Design

Because the 3 phase motor is synchronous, under normal operation, the step response would tend to go to its reference value without the need of a controller however, due to any vibrations and feedback signals (back EMF etc.) as well as any other external disturbances, a PID controller must be implemented in order to ensure a fast and steady velocity/position control. The below figure is a simple and standard control architecture used to control this motor.

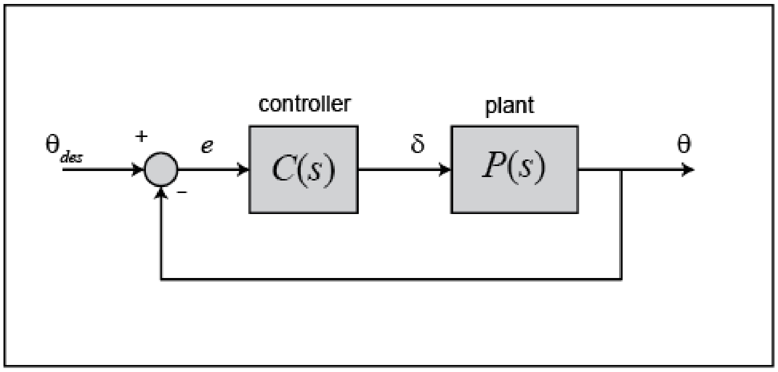


Figure 9: *Controller loop used for the three phase motor.*

4.1. Controller Form and Design Requirements

The controller function, C(s), which follows the general structure of a standard PID controller, and the plant function, P(s), given by the manufacturer, are shown below in equations (1) and (2). Here, J_{eq} is given to be 1.1e^{-6} kg/m^2 [1].

$$C(s) = K_p\left(1 + \left(\frac{s}{K_d}\right) + \left(\frac{K_i}{s}\right)\right) \quad \textbf{(1)}$$

$$P(s) = \frac{1}{J_{eq}s^2} \quad \textbf{(2)}$$

Where K_p, K_d, and K_i are the proportional, derivative and integral gains of the controller. The design requirements for this controller are (ideally) as follows:

- Rise time <= 0.5 milliseconds
- Settling time <= 1 millisecond (to keep up with the update rate of the system)
- Percent Overshoot <= 2

4.2. Theoretical Calculations

The first stage of the PID controller design (i.e. finding the correct gains) involves the theoretical calculation of the gains by use of MATLAB's siso tool toolbox. The theoretical gains assume that the motor and all other modules interfacing the motor with our MCU are ideal (i.e. no external noise, disturbances etc.). This idealized calculation serves as a benchmark or starting point from which we can tune further to compensate for a non-ideal model.

The figure below shows the root locus, bode plot and step response of an uncontrolled P(S) transfer function.

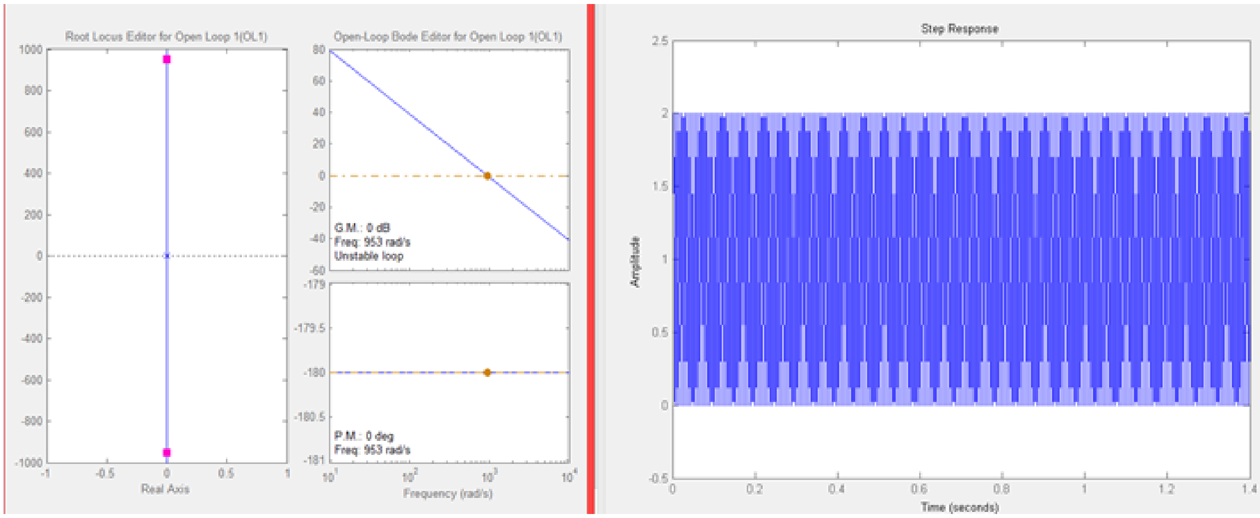


Figure 10: *Response of our system without C(s)*

A controller was then added to the model architecture and tuned by using a "trial and error" approach of changing the locations of the poles and zeroes of the controller in the root locus until the appropriate design requirements are met. In this case, the following PID gains were found to meet the theoretical design requirements:

$$K_p = 0.00591,\ K_d = 0.0141,\ K_i = 9.93E\text{-}5$$

As expected, due to the motor's synchronous nature, the gains required to control it are quite small. The improved analysis plots are shown below in Figure 11 with the controller included:

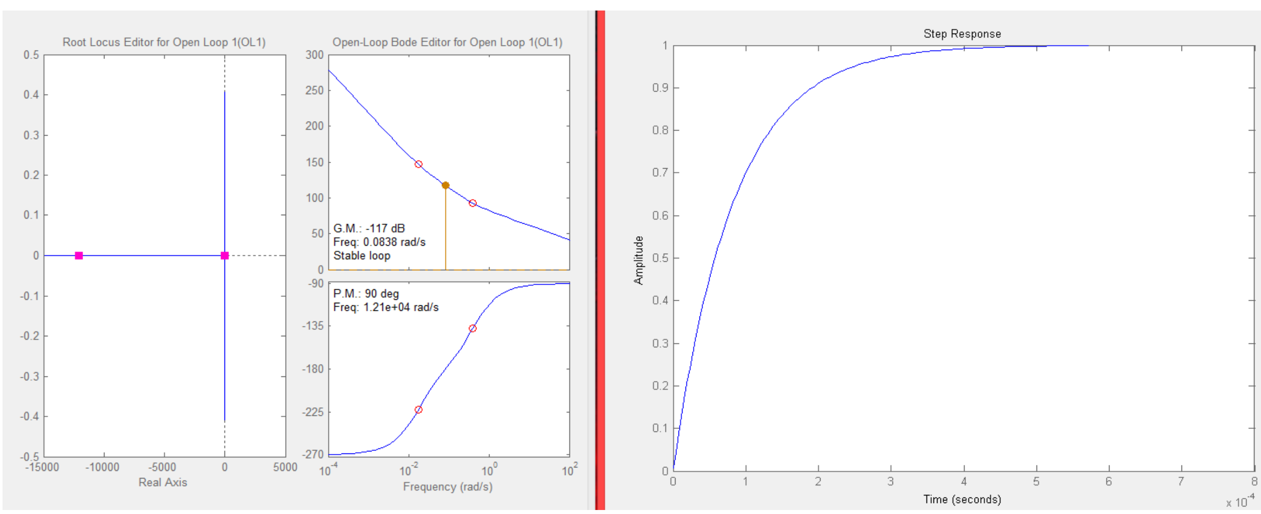


Figure 11: *Design and response of the system with the PID controller*

4.3. Discretization of Controller

The trapezoidal method was used in order to discretize the C(S) transfer function where U represents the controller output and E represents the error signal (or controller input) =

Reference – Motor Encoder reading. The equation for the discretization is shown below in Equation (3).

$$U_k = U_{k-1} + (K_p + K_i + K_d) * E_k - (K_p - 2 * K_d) * E_{k-1} + K_d * E_{k-2} \quad \textbf{(3)}$$

4.4. The PID Module

In order to be implemented in C to be recognized by the MCU, a PID module was created that follows the following structure (see code provided for details):

PID
Input Signal: float err //error signal = reference for velocity/position - velocity/position sensed by encoder //reference signals are set by the MPU 6050 gyroscope Output Signal: float PID_Out //output of controller calculation funciton

Parameters:
float KP;
float KD;
float KI;
float c[3];
float Ts;
float upplim; //Upper Limit of PID output to prevent integral windup
float lowlim; //Lower Limit of PID output to prevent integral windup
float InputBuffer[3]; //stores the current and previous 2 samples signals for calculation
float OutputBuffer[2]; //stores the current and previous controller output samples

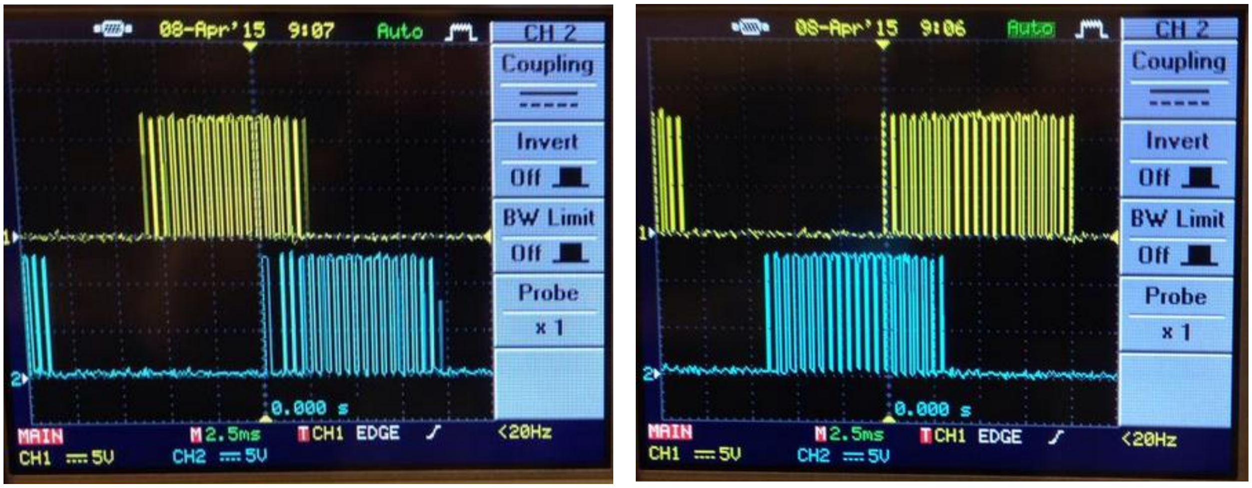
Functions:
void PID_Init(PID_TypeDef PIDin)//Initializes and sets all PID parameters
float CalcPID_Out(PID_TypeDef PIDin, float err)//uses the discretized controller function and anti windup saturation in order to calculate the PID controller output

Table 2: *Design class for the PID controller*

5. Experimental Results

5.1. 3 Phase Sine Waves

The three sine phases generated are shown below in Figure 12 (a) and (b). These figures confirm that we are generated three sine waves that are 120 degrees out of phase, as desired.

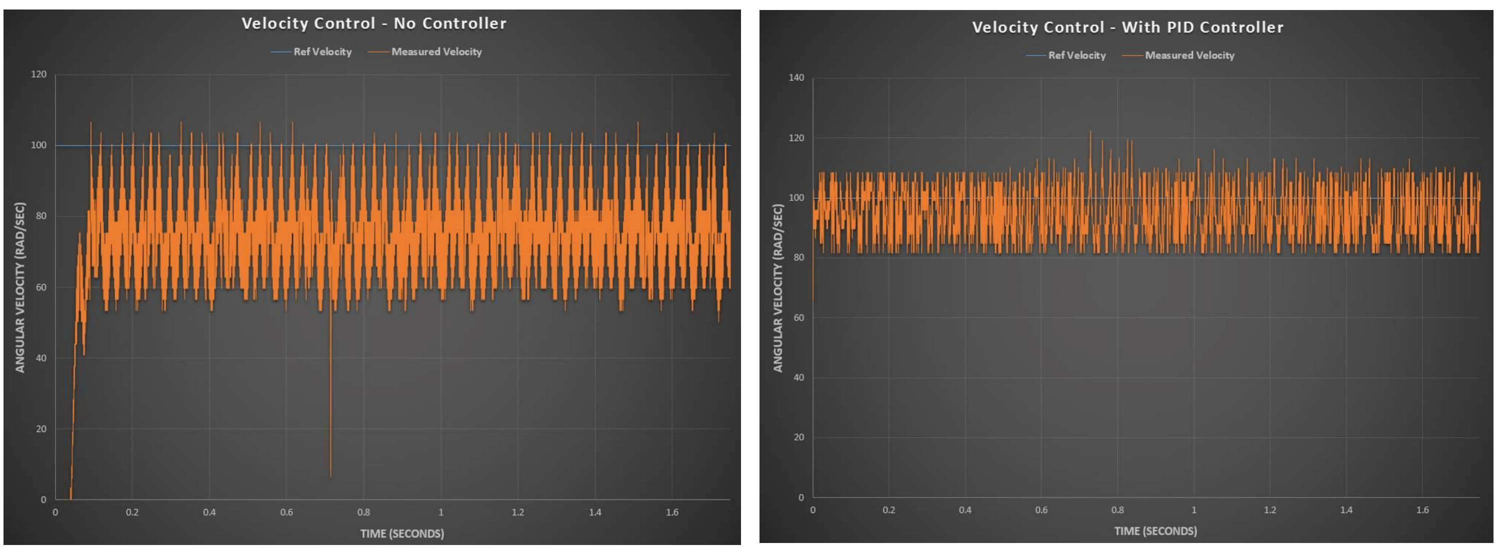


Phase 1 (yellow) vs Phase 3 (blue) ***Phase 1 (yellow) vs Phase 3 (blue)***

Figure 12: *Generated sine signals to control the BLDC motor*

5.2. Controller Output

In order to test the effectiveness of the controller, the step response of the system was gathered with and without the controller. In both cases, the desired velocity was set to be 100 radians per second. The results are plotted below in Figure 13 (a) and (b).



(a) Velocity response, no controller ***(b) Velocity response, with PID Controller***

Figure 13: *Velocity response of the system with and without the PID controller*

It is clear from the above figures that the addition of our PID did benefit the system. The response with the controller had a faster rise time, less percentage overshoot, and a more accurate desired value.

6. Conclusion

In this project, we were successfully able to control the velocity of a brushless DC 3 phase motor with a gyroscope. In order to increase the accuracy of the velocity response, a controller was designed, discretized, and integrated into our C code. In order to verify the controller actually helped, the response of the system with and without the controller was observed, and it was clear that the addition of a PID controller helped our system.

7. References

[1] Maxon Motors, “EC-max 30 datasheet”. 2014. [Online]. Available: http://www.maxonmotor.com/medias/sys_master/root/8813561249822/14-204-EN.pdf

[2] MC9S12DP256 Advance Information, Freescale Semiconductor, 2001.

[3] InvenSense, “MPU-3050 Triple Axis Gyroscope with Embedded Digital Motion Processor”, invensense.com. [Online]. Available: http://www.invensense.com/mems/gyro/mpu3050.html

[4] Performance Motion Devices, “Brushless DC Hall Effect Sensor Configuration”. [Online]. Available: http://www.pmdcorp.com/downloads/app_notes/BrushlessSensorConfig.pdf

[5] E. Park. MSE 310. Class Lecture, Topic: "Actuators." Simon Fraser University, Surrey, British Columbia, Oct. 31, 2013.

[6] Wikidot, “Sensorless BLDC and 8051-based Microcontroller”. [Online]. Available: http://bldc.wikidot.com/bldc-and-8051

[7] ST Microelectronics, “L6234 Three Phase Motor Driver”. [Online]. Available: http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/application_note/CD00004062.pdf